q2.27

dm

2/1/2022

Please run the code chunk below to load the gamble and matrix power functions first! The problem and solution starts on next page.

```
# gamblersruin.R
# Example 1.11

# gamble(k, n, p)
  #   k: Gambler's initial state
  #   n: Gambler plays until either $n or Ruin
  #   p: Probability of winning $1 at each play
  #   Function returns 1 if gambler is eventually ruined
  #                    returns 0 if gambler eventually wins $n

gamble <- function(k,n,p) {
    stake <- k
    while (stake > 0 & stake < n) {
        bet <- sample(c(-1,1),1,prob=c(1-p,p))
        stake <- stake + bet
    }
    if (stake == 0) return(1) else return(0)
    }

#k <- 10
#n <-  40
#p <- 1/2
#trials <- 1000
#simlist <- replicate(trials, gamble(k, n, p))
#mean(simlist) # Estimate of probability that gambler is ruined
# For p = 0.5, exact probability is (n-k)/n

##### Matrix powers ##############################
# matrixpower(mat,k) mat^k
#
matrixpower <- function(mat,k) {
    if (k == 0) return (diag(dim(mat)[1]))
    if (k == 1) return(mat)
    if (k > 1) return( mat %*% matrixpower(mat, k-1))
 }
```

R Markdown

2.27 R : See gamblersruin.R. Simulate gambler's ruin for a gambler with initial stake \$2, playing a fair game. (a) Estimate the probability that the gambler is ruined before he wins \$5. (b) Construct the transition matrix for the associated Markov chain. Estimate the desired probability in (a) by taking high matrix powers. (c) Compare your results with the exact probability.

ANS:

```
######## part (a) #########

n_trials <- 100000
# initial stake = 2, gambler ruined before he wins 5,
# fair game => prob = 0.5
simulation_list <- replicate(n_trials,gamble(2,5,0.5))
# finding mean to "estimate"
mean(simulation_list)
```

```
## [1] 0.60006
```

```
######## part (b) ########
p_matrix <- matrix(c(1,1/2,0,0,0,0,
                     0,0,1/2,0,0,0,
                     0,1/2,0,1/2,0,0,
                     0,0,1/2,0,1/2,0,
                     0,0,0,1/2,0,0,
                     0,0,0,0,1/2,1),nrow=6)
rownames(p_matrix) <- 0:5
colnames(p_matrix) <- 0:5
matrixpower(p_matrix,100)
```

```
##     0            1            2            3            4   5
## 0 1.0 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.0
## 1 0.8 1.726998e-10 0.000000e+00 2.794341e-10 0.000000e+00 0.2
## 2 0.6 0.000000e+00 4.521339e-10 0.000000e+00 2.794341e-10 0.4
## 3 0.4 2.794341e-10 0.000000e+00 4.521339e-10 0.000000e+00 0.6
## 4 0.2 0.000000e+00 2.794341e-10 0.000000e+00 1.726998e-10 0.8
## 5 0.0 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.0
```

```
print("As we can see in the resultant matrix, the requested probability is 0.6")
```

```
## [1] "As we can see in the resultant matrix, the requested probability is 0.6"
```

```
########## part (c) #########
# desired probability can be given as (n-k)/n
print((5-2)/5)
```

```
## [1] 0.6
```